Topic Tracking in a News Stream

J.P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt

Dragon Systems, Inc.
320 Nevada Street
Newton, MA 02460

ABSTRACT

In this paper we describe a Topic Tracking system based on unigram models, submitted by Dragon Systems in the December 1998 Topic Detection and Tracking (TDT) Evaluation. We focus on the most recent developments, including improvements in the smoothing of sparse unigram models, a better discriminator, and the implementation of unsupervised adaptation. We give results on the default test conditions, namely, tracking in newswire and automatically recognized broadcast given four story samples, as well as several variations: one story sample, automatically recognized broadcast only, and automatically recognized broadcast with automatically determined story boundaries. Finally, we show the effect of interpolating this system with Dragon's other tracking system based on a Beta-Binomial model.

1. INTRODUCTION

The DARPA Topic Detection and Tracking (TDT) program is concerned with the development of information processing technology that can applied to large streams of data, such as newswire and broadcast news [1]. To facilitate research on the TDT tasks, the Linguistic Data Consortium (LDC) has created the *TDT2 corpus*, a collection of newswire and transcribed broadcasts from a variety of sources covering January through June 1998. The broadcasts (approximately 800 hours) were transcribed both automatically and in closed-caption, the automatic version generated using a modification of Dragon's 1997 Hub-4 recognizer [2]. A feature of this corpus, key to this research, is that each story has been labeled with a binary decision as to its relevance to each of 100 topics.

In the Tracking task (a variation of the filtering task in information retrieval), a system is supplied with a few examples of stories on a particular topic of interest, and is expected to automatically find subsequent examples in the stream. Specifically, a system is given as training material the first N_t examples in the evaluation corpus of stories on a particular topic (the *topic training stories*), plus all off-topic stories in the evaluation corpus prior to the last training example (*off-topic training stories*), plus all stories prior to the evaluation corpus (*background data*), and asked to return judgments on all remaining evaluation stories.

In this paper we will describe the second incarnation of a tracking system which uses standard language modeling techniques (in particular, unigram statistics) to measure document similarity. As in our earlier work [3, 4], this system is based on a simple classifier:

- Score an incoming story against a topic unigram language model built from the topic training stories
- Score the story against a discriminator language model built from the background data
- Output the difference between these scores as a relevance value, or threshold this difference to generate a decision

One of the key ways in which this system is different from its earlier incarnations is in the way we smooth the extremely sparse topic unigram models that arise from the topic training stories. We have improved the targeting procedure and introduced a variation on linear discounting that has significantly improved performance. These techniques are described in Section 2.

The nature of the TDT2 corpus makes it likely that, for any given evaluation topic, the data from which we build our multiple discriminators (the TDT2 training and development sets, or January–April 1998 data) is "contaminated" with on-topic material. For this reason we are now careful to filter such material in the construction of the discriminator models. In addition, one of these models is now targeted specifically to the tracked topic to better discriminate on-topic and close-to-topic stories. The discriminator is described in Section 3.

Other techniques we have implemented, including unsupervised adaptation on high-scoring test stories, is described in Section 4. Evaluation (and some post-evaluation) results, including the effect of interpolating the unigram tracker with Dragon's Beta-Binomial tracker, are presented in Section 5.

2. SMOOTHING OF THE TOPIC MODELS

Our approach to the smoothing problem has focused on the use of *targeting,* in which we take a large number of language models built from the background material, find the mixture that best approximates the sparse model, and use this mixture as a smoothing distribution.

More concretely, given a sparse topic unigram model $t(w_n)$ built from the topic training data, and a set of *background models* $b^{(i)}(w_n)$, we find the best mixture

$$b(w_n) = \sum_i \lambda^{(i)} b^{(i)}(w_n) \ , \ \sum_i \lambda^{(i)} = 1 \ ,$$

such that the Kullback-Leibler distance between $t(w_n)$ and $b(w_n)$,

$$d = \sum_n t(w_n) \log \frac{t(w_n)}{b(w_n)} \ ,$$

is minimized. This leads to an implicit equation for the $\lambda^{(i)}$:

$$\lambda^{(i)} = \sum_n \frac{t(w_n) \lambda^{(i)} b^{(i)}(w_n)}{\sum_j \lambda^{(j)} b^{(j)}(w_n)} \ ,$$

which is easily solved by iteration.

In earlier versions of our system we targeted the topic unigram model against unigram models derived from clusters of stories from the background data (typically about 100 models). In this investigation, we targeted against the unigram models associated with the individual background stories (15,000–50,000 models—although for reasons having to do with the discriminator, some of the background stories were filtered out first; see Section 3). Our motivation is that a mixture based on documents can select background data more like the topic training data, and therefore generalize that data in a more realistic way compared to a mixture based on coarse clusters.

One problem that can result when targeting to individual stories is the assignment of a large proportion of the mixture probability to a small number of stories, yielding a mixture distribution which is itself sparse. To measure the sparseness of the mixture (recall it is a *probability* distribution built from a large number of components, and so may not actually contain zeros), we assign it a total count B according to

$$B = \exp\Bigl(\sum_i \lambda^{(i)} \log \frac{c^{(i)}}{\lambda^{(i)}} \Bigr) \ ,$$

where $c^{(i)}$ is the total count of background story i. (To understand this formula, consider the case in which all background stories have the same total count c. The expression for B then reduces to

$$B = c \exp\Bigl(- \sum_i \lambda^{(i)} \log \lambda^{(i)} \Bigr) \ ,$$

or c times the perplexity of the mixture weight distribution. Roughly speaking, this perplexity is the number of *stories* over which the mixture is distributed, so B represents the number of *counts* over which it is distributed.) Given a total count B, the mixture distribution is converted to counts and smoothed.

In our 1997 system we smoothed the mixture and topic unigram models by absolute discounting [5] followed by backoff to a smoothing distribution [6]. However, we have observed that in very sparse models, absolute discounting appears to be insufficiently aggressive at redistributing probability. For that reason we switched in the 1998 system to linear discounting (in which the amount discounted from each count is proportional to the count), with the linear discount parameter determined by requiring that the smoothed distribution have a specified internal perplexity, large enough to guarantee that the smoothed model has its counts distributed over a large number of words. Using this method, the targeted mixture model associated with each topic was smoothed with the global background distribution, and the topic model was then smoothed with the smoothed targeted mixture model.

3. THE DISCRIMINATOR

The discriminator for Dragon's previous system consisted of a large number of unigram models derived by automatically clustering the background material. For any given test story, the best scoring model from this set is the one chosen to compare to the topic model. The advantage of such a system is that an off-topic test story will tend to score well in at least one of the clusters, allowing it to be easily distinguished from the tracked topic.

What this system does not handle as well is the case of the off-topic story that shares features with the tracked topic. Consider, for example, the problem of distinguishing a story on a tobacco lawsuit brought by an individual, from a topic concerning the national tobacco settlement. Unless there is a background cluster concerned with tobacco lawsuits, the story will likely get a good tracking score.

To address this problem, the new system includes in the discriminator a model that is designed to be “close” to the topic model without actually containing topic training data. It is expected that this model will be the best scoring of the discriminator models for on-topic and close-to-topic stories.

The obvious candidate for a “close” model is the targeted mixture model built to smooth the topic model. However, in order for the mixture model to work properly as a discriminator, it is crucial that the background from which it is derived be free of any on-topic material. Therefore, before doing the targeting described in the previous section, we build a rudimentary tracker and “track” the background stories. Any stories that score too well are presumed to be on topic, and are discarded. (One could use these high-scoring stories to supplement the topic training material, but this was not done in this investigation.) Targeting is then done only against the remaining stories.

Although we are careful to remove on-topic material from the background before targeting the mixture model, we do not remove it prior to producing the background clusters from which the other unigram models in the discriminator are derived (this would have required a clustering run in every tracking experiment, which is too costly). This means that for a given topic, one or more of the clusters may be contaminated with on-topic data. To correct for this, the set of cluster models is filtered to remove any that the targeted mixture model fails to outscore by a certain threshold on the topic training material.

4. OTHER TECHNIQUES

The tracker includes a mechanism for unsupervised adaptation on incoming stories that are highly likely to be on topic. If a story comes in that scores higher than a specified threshold, this story is added to the set of topic training stories, and the entire build procedure is rerun. This includes:

- A preliminary tracking of the background to remove on-topic material
- Targeting a new mixture model to use as a smoothing distribution and as a discriminator
- Smoothing the topic model and the targeted mixture model
- Filtering the background clusters to remove any that may be contaminated with on-topic material

Tracking then continues on the next available test story. Unsupervised adaptation had a small positive effect on performance.

Other techniques that were tried and rejected because they had little of no effect included a penalty on unusually short test stories and a *time penalty* that caused test stories to become less likely to be considered on topic the further into the corpus they appeared. The time penalty was a successful feature of our previous system, presumably because news on a topic tends to die out over time; its lack of utility here is probably due to the shorter time frame of the evaluation corpus compared to last year (two months vs. one year).

5. RESULTS

In our development system, each topic unigram model was targeted against approximately 15,000 background stories from the TDT2 January–February data. A stop list of about 100 common words was applied before targeting. The targeted mixture model associated with each topic was smoothed with the global background distribution to an internal perplexity of 1500 (determined by tuning), and the topic model was then smoothed with the smoothed targeted mixture model, also to an internal perplexity of 1500. The discriminator consisted of the targeted mixture model and 100 automatically derived clusters of the background data. The development test material consists of the TDT2 March–April data.

Figure 1 shows a comparison of our 1998 and 1997 systems on the development test set, running under the default evaluation conditions: tracking with four story samples ($N_t = 4$) in newswire (NWT) and automatically recognized broadcast (ASR). The detection-error tradeoff (DET) plots are generated by pooling the output of the tracker from the different topic runs and sweeping a decision threshold through the story relevance scores. A high threshold causes only very high scoring stories to be reported as on-topic, which tends to produce to high miss rate for the topic, but a low false-alarm rate. Conversely, a low threshold results in most on-topic stories being identified, and hence a low miss rate, but the false-alarm rate rises. The fact that the 1998 plot is mostly well inside the 1997 plot indicates that the 1998 system is substantially improved over the old system.

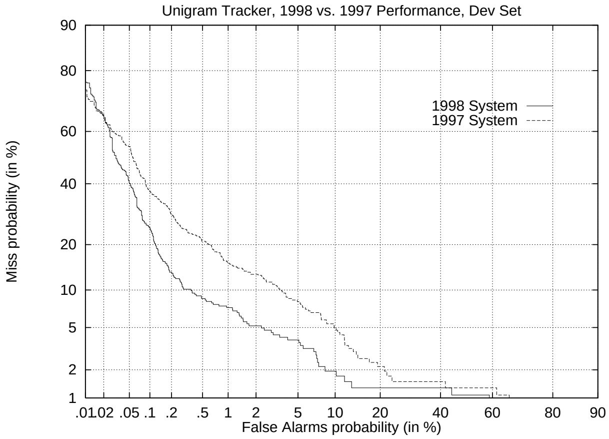

Figure 1: Comparison of 1997 and 1998 tracking results, Unigram tracker.

The evaluation system is identical to the development system except that the background was taken to be the approximately 49,000 stories that comprise the TDT2 January–April data. The discriminator, once again, consisted of the targeted mixture model and 100 automatically derived clusters of the background data. The evaluation test data covers the May–June portion of the TDT2 corpus.

Our performance on the evaluation test data for the default evaluation conditions is shown on Figure 2, along with a contrast showing the effect of reducing the number of topic training samples to $N_t = 1$. (This is somewhat of an unfair comparison, as the system parameters were tuned for performance at $N_t = 4$.)

One goal of the 1998 evaluation was to see what effect certain kinds of errors in the input have on performance. Figure 3 presents two comparisons: first, the performance

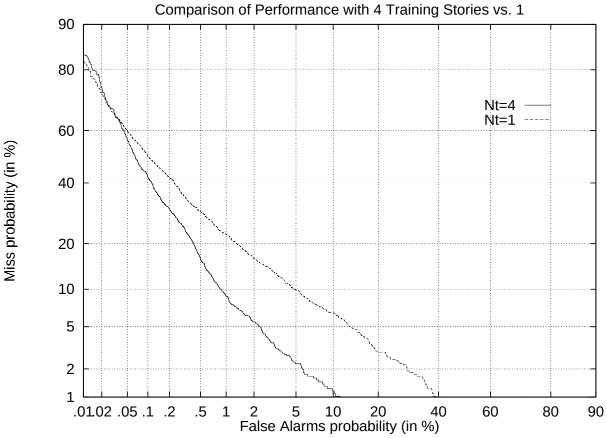

Figure 2: Effect of reduced training.

on newswire and automatically recognized broadcast compared to automatically recognized broadcast only, and second, the performance on automatically recognized broadcast compared to the same data with story boundaries determined automatically by Dragon's HMM segmenter [7].

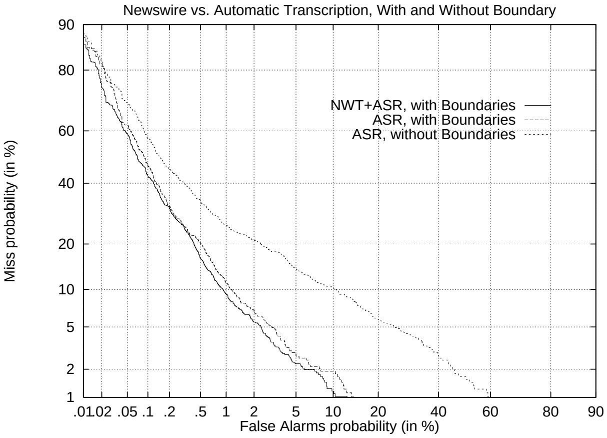

Figure 3: Newswire vs. automatic transcription, story boundaries given vs. not given.

Figure 3 shows that there is almost no degradation in performance associated with the automatically recognized material, despite a word error rate on the order of 30%. On the other hand, there is a noticeable loss of performance when story boundaries are determined by machine; this is discussed further in [7].

Dragon submitted two tracking systems for evaluation, the one described here and another based on a Beta-Binomial model [8]. Figure 4 shows the result of interpolating the output of the two trackers, on evaluation data. Performance was fairly insensitive to the tuning of the mixture, which was set to 50-50 based on results on the development data. The interpolated system outperforms both of its components.

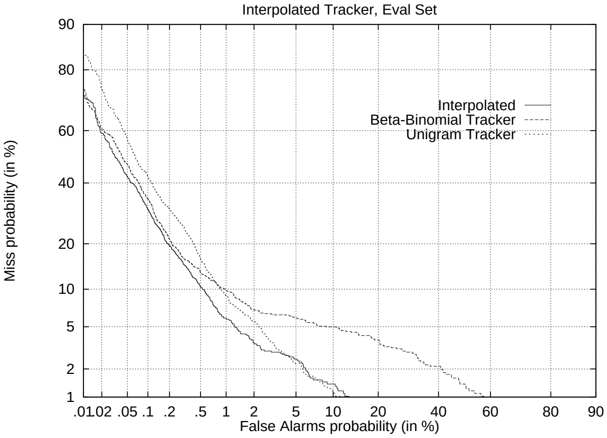

Figure 4: Interpolation of Beta-Binomial tracker and Unigram tracker.

Given a decision threshold on the tracking scores, the evaluation provides a single component metric, C_{track}, for measuring system performance (smaller values are better). For the unigram tracker described here, the value of this metric on the evaluation data was $C_{track} = 0.0079$, and for the Beta-Binomial model it was $C_{track} = 0.0071$. The interpolated model achieved the value $C_{track} = 0.0062$. All decision thresholds were tuned on the development data.

References

1. J. Allan, J. Carbonell, G. Doddington, J.P. Yamron, and Y. Yang, "Topic Detection and Tracking Pilot Study: Final Report," *Proceedings of Broadcast News Transcription and Understanding Workshop,* Lansdowne, Virginia, February 1998.
2. L. Gillick, Y. Ito, L. Manganaro, M. Newman, F. Scattone, S. Wegmann, J.P. Yamron, and P. Zhan, "Dragon Systems' Automatic Transcription of New TDT Corpus," *Proceedings of Broadcast News Transcription and Understanding Workshop,* Lansdowne, Virginia, February 1998.
3. J.P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. van Mulbregt, "A Hidden Markov Model Approach to Text Segmentation and Event Tracking," *Proceedings ICASSP-98,* Seattle, May 1998.
4. P. van Mulbregt, J.P. Yamron, I. Carp, L. Gillick, and S. Lowe, "Text Segmentation and Topic Tracking on Broadcast News via a Hidden Markov Model Approach." *Proceedings ICSLP-98,* Sydney, December 1998.
5. H. Ney, U. Essen, and R. Kneser, "On Structuring Probabilistic Dependences in Stochastic Language Modelling," *Computer Speech and Language,* 8:1–38, 1994.
6. S. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," *IEEE Transactions on Acoustics, Speech and Signal Processing,* ASSP-35(3):400–401, March 1987.
7. P. van Mulbregt, I. Carp, L. Gillick, S. Lowe, and J.P. Yamron, "Segmentation of Automatically Transcribed Broadcast News Text," elsewhere in this *Proceedings,* February 1999.
8. S. Lowe, "The Beta-Binomial Mixture Model and Its Application to TDT Tracking and Detection," elsewhere in this *Proceedings,* February 1999.